Efficiency and Effectiveness of Data Warehousing: A Case Study

Carlo DELL'AQUILA, Ezio LEFONS, and Filippo TANGORRA
Dipartimento di Informatica
Università di Bari
via Orabona 4, 70125 Bari
ITALY
{dell'aquila, lefons, tangorra}@di.uniba.it

Abstract: - The decisional analysis is a complex, iterative, and exploratory process based on a sequence of queries issued against the data warehouse. The *efficiency* of the decisional activity depends on the quality of the stored data in the data warehouse, while its *effectiveness* is related to the analysis tools.
In this paper, we discuss both these aspects in the context of a service company, with the aim of analyzing the trend of railway booking data. In particular, the back-end and front-end software solutions of data warehousing implementation are presented, which permit to met the customer needs while preserving the desired requirements of efficiency and effectiveness.

Key Words: - Data warehousing, back-end solutions, front-end solutions, railway travel booking, train transportation analysis.

1 Introduction

Data warehouses have gained an increasing prominent role in supporting deep analysis and strategic planning. In fact, they are very large repositories that integrate data coming from operational databases of several enterprise sectors for decisional analysis. Data warehouses are implemented like databases and they are not designed for online transaction processing, but for read-only ad hoc analysis for business data [2, 4, 10, 11, 14].
Actually, on-line analytic processing (OLAP) systems and decision-support (DS) systems run on data warehousing environments built on the relational database technology (ROLAP servers). Typically, OLAP and DS applications involve the computation of summary data and the execution of aggregate queries [9].
One of the main issues of data warehousing is to store non-volatile historical data for query purposes, since they are not data archival systems. Usually, a data archival system takes data from a single operational source system and, even if it stores data from several systems, there is low or no integration of data from the heterogeneous source systems. Cross-system data integration is a relevant characteristics of data warehousing: data loading in a data warehouse is often a complex process involving data cleaning and transformation. The source environment of a data archival system consists often of source systems that have been put out of use, while data warehouses take data from live operational source systems. Therefore, quality of data in the warehouse is a primary goal in order to respond efficiently and accurately to unforeseeable queries at any moment of time.
On the other hand, data analysis is an iterative and exploratory process in nature, and it is characterized by a sequence of queries, in which each successive query to the data warehouse could being influenced by the results of the previous ones.
Generally, statistical and data-mining procedures do not require complex query languages to satisfy the user needs. In fact, OLAP and DS applications involve the computation of summary data and the execution of aggregate queries (queries requiring data aggregation and grouping such as the SQL operators COUNT, SUM, MIN, MAX, and AVG with GROUP BY / HAVING clauses). But, if one considers to repeat the analysis process several times, sifting through many results, it is crucial to choose suitable tools in order to get effectiveness in the data visualization, statistics, data mining, and knowledge discovery activities.
The reference context considered in this paper, is the Italian railway data warehousing system for national booking data analyses. Here, we present the back-end and front-end solutions of the data warehousing implementation, which permitted us to meet the customer needs while preserving the desired requirements for efficiency and effectiveness.

2 Data Warehousing system architecture

The architecture overview of the application reference context—the Italian railway system for the booking process [5]—is depicted in Figure 1.
The overall system comprises the usual levels of data flow corresponding to the sequence of steps for

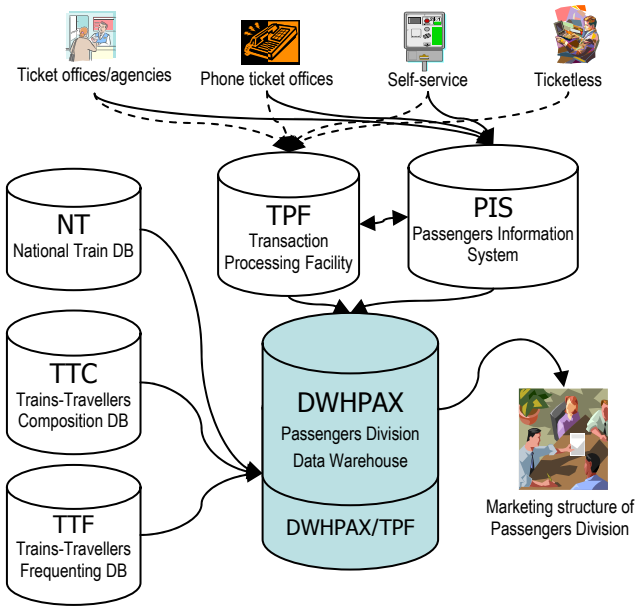


Fig. 1. Italian railway data warehouse architecture.

adapting the data to the decision-maker needs [3, 7-10]. The warehouse is structured on two layers: Operational Data Store (ODS) and Enterprisewide Data Warehouse (EDW). The ODS stores tactical data from production systems that are subject-oriented and integrated to address operational needs. The EDW stores data from all subject areas within the business for analysis by end users. Its scope is the entire business and all operational aspects within the business.

The source data flows dealt with by Transaction Processing Facility (TPF, *cf.,* Figure 1) come from the Sale System of the operational Passengers Information System (PIS) and from the set of information occurring in the booking analysis realization. Besides PIS/TPF, the other operational systems involved are: the National Train database (NT), which contains the train route and railway kilometres; the Trains-Travellers Composition database (TTC), which contains saleable delivered services, associated trains and antenna trains (coupling/release); and the Trains-Travellers Frequenting database (TTF), which contains the train registry.

3 Hardware architecture

The prototype consists of the following equipment:

- IBM host calculus capacity for TPF data acquisition, storing, validation and loading,
- centralized Windows NT server for TTC and NT data acquisition,
- centralized UNIX server, hosting the data warehouse and containing Maintenance Areas, Train Production and Commercial Data,
- centralized Windows 2000 professional server, hosting the Web server and the Microstrategy's Intelligent Server (the reporting engine), and
- user workstations, linked to these servers by the Geographical Network WAN to use the application.

The system takes into account:

- scalability of the architectural solution, providing support for the database growth both in size and in amount of users.
- Data integrity safety in case of system crashes or high system availability, through fault-tolerance mechanisms.
- Quick backup and recovery operations with the aim of avoiding the obstruction of other system activities, and providing accorded restore times.

4 Software solutions

As concerns the main software solutions adopted in implementing the data warehouse and related to the analytical user requirements, we consider in detail the following aspects:

- *Back-end solutions*:
  - The technical offer for the re-engineering of source TPF database.
  - The ETL process of the daily data to update the ODS (Operational Data Store) and the EDW (Enterprisewide Data Warehouse) layers.
- *Front-end solutions*:
  - The customer requests.
  - The reports in Microstrategy on the basis of the analytic user requirements.

4.1 Back-end solutions

Acquisition and population functions work on several distinct elaboration environments: client/server environments, where TTC and NT databases are resident; data warehouse server environment, interacting with other systems, running on a UNIX machine where the Passengers Division warehouse database resides and where the Frequenting database, from which the Train Registry will be taken, currently resides. The database is implemented with Oracle 9i [1, 6, 12, 13].

The TTC environment produces, through SQL-Server procedures, the flat files that will be loaded in the Data Warehouse server environment through FTP procedures. Similar procedures load the NT data in the Data Warehouse server. These procedures are designed and realized with the double purpose of (1) retrieving all the previous data about associated trains which lie in the interested period and (2) gradually reducing and

discharging of the Passengers Division host, and they replace the current host procedures.

At the same time, all information about Train Registry will be moved from TTF database to the data warehouse server. These data, together with those coming from other source systems, will be managed using procedures written in: (a) SQL*Loader, for loading on the ODS layer, and (b) PL/SQL, for loading on the EDW and Data Mart layers.

The other data that acquisition and population functions need will be directly taken from the Data Warehouse database.

Information about Delivered Services, Associated Trains and Antenna Trains is loaded from TTC source system (*see,* Figures 2-3). Train Routes, with respective validity dates and kilometre number, are taken from the NT source system. Information about Train Registry is present on DWHPAX in the TTF instances.

ZAWARE01.BATCH.DWH.COMMTPF.TTCGIO FILE	
FIELD NAME	**DESCRIPTION**
DAT_TRE	Train Date
NUM_TRE	Train Number
SUF_TRE	Train Suffix
COD_RET_INI_TRA	TTC Elementary Route Source Station Railway Code
COD_STA_INI_TRA	TTC Elementary Route Source Station Code
COD_RET_FIN_TRA	TTC Elementary Route Destination Station Railway Code
COD_STA_FIN_TRA	TTC Elementary Route Destination Station Code
PSI_PRI_OFF	First Class Seats Delivered Total Number
PSI_SEC_OFF	Second Class Seats Delivered Total Number
CUC_PRI_OFF	First Class Couchettes Delivered Total Number
CUC_SEC_OFF	Second Class Couchettes Delivered Total Number
CUC_CMF_OFF	Couchettes Delivered Total Number
PSI_LTT_OFF	Beds Delivered Total Number
PSI_TAA_OFF	Following Car Places Delivered Total Number

Fig. 2. Delivered Services (TTC DB).

ZAWARE01.BATCH.DWH.COMMTPF.TTCNTN FILE	
FIELD NAME	**DESCRIPTION**
DAT_TRE	Antenna Train Date
NUM_TRE	Antenna Train Number
DAT_TRE_PRN	Main Train Date
NUM_TRE_PRN	Main Train Number
COD_STA_SGA	Release Station Code
COD_RET_DES	Main Train Destination Railway Code
COD_STA_DES	Main Train Destination Station Code
COD_STA_TRT	Transit Station Code

Fig. 3. Antenna Trains (TTC DB).

TPF flow data (coming from PIS/TPF) have the following meanings ("S/T" stands for "source/target station"):

- *Train Date:* train leaving date
- *Train Number:* train number
- *Train Railway:* railway the train belongs to
- *"From,, Station Code:* source station code of the booking
- *"From,, Railway Station Code:* source railway code of the booking
- *"To,, Station Code:* destination station code of the booking
- *"To,, Railway Station Code:* destination railway code of the booking
- *First Class Seats Total:* total number of booked travels for date, train, S/T in first class seats
- *Second Class Seats Total:* total number of booked travels for date, train, S/T in second class seats
- *First Class Couchettes Total:* total number of booked travels for date, train, S/T in first class couchettes
- *Second Class Couchettes Total:* total number of booked travels for date, train, S/T in second class couchettes
- *Comfort Couchettes Total:* total number of booked travels for date, train, S/T in comfort couchettes
- *Beds Total:* total number of booked travels for date, train, S/T in beds
- *Following Cars Total:* total number of booked travels for date, train, S/T in following cars
- *Booking Type:* "Y/N" Flag.

TPF files are structured as follows (#bytes: description):

01-10: Train leaving date
11-16: Train number
17-19: Train railway
20-24: CCR station *From*
25-27: Railway station *From*
28-32: CCR station *To*
33-35: Railway station *To*
36-39: First class seats booked
40-43: Second class seats booked
44-47: 4-place couchettes booked
48-51: 6-place couchettes booked
52-55: Comfort couchettes booked
56-59: WL beds booked
60-63: Cars followed places booked
64-64: Booking type.

Data Pre-calculation

Since every TTC and TPF records are associated with one generally composite source/target, and composition train and delivered services do not change, for each composite source/target, delivered services (TTC) and booking (TPF) will be assigned to every elementary route. This calculation, preparatory to the mapping, occurs in the TTC/NT environment and it is the input for the first loading in the operational data store layer, as shown in Figure 4.

ZAWARE01.BATCH.DWH.COMMTPF.TTCGIO.PREEL FILE	
FIELD NAME	DESCRIPTION
DAT_TRE	Train Date
NUM_TRE	Train Number
SUF_TRE	Train Suffix
PRG_TRA	Elementary Route Number of the Train Route Sort
COD_RET_INI_TRA	Elementary Route Leaving Railway Code
COD_STA_INI_TRA	Elementary Route Leaving Station Code
COD_RET_FIN_TRA	Elementary Route Final Railway Code
COD_STA_FIN_TRA	Elementary Route Final Station Code
PSI_PRI_OFF	First Class Seats Delivered Total Number in the Elementary Route Initial Station
PSI_SEC_OFF	Second Class Seats Delivered Total Number in the Elementary Route Initial Station
CUC_PRI_OFF	First Class Couchettes Delivered Total Number in the Elementary Route Initial Station
CUC_SEC_OFF	Second Class Couchettes Delivered Total Number in the Elementary Route Initial Station
CUC_CMF_OFF	Couchettes Delivered Total Number in the Elementary Route Initial Station
PSI_LTT_OFF	Beds Delivered Total Number in the Elementary Route Initial Station
PSI_TAA_OFF	Following Car Places Delivered Total Number in the Elementary Route Initial Station
TRE_KMT_RET	Railway Kilometre Train in the Elementary Route

Fig. 4. Elementary Route Data pre-calculation.

EDW Population

The Train Registry (*see,* Figure 5) is loaded from the TTF one, where special trains are defined: it will be merged with the PIS/TPF files, which are not known by TTF, and thus not classifiable. This registry is daily updated in order to consider variations occurred (train number change, new special trains, registry modifications, *etc.*). In this way, previously not classified trains will be completed with the correct information (TTF classification, relation, S/T, *etc.*) as soon as it is available.

TPF booking and TPF-120 booking files will be re-calculated in order to bring associated train booking to the corresponding master train, with attention to the fact

CMM_EDW_TPF_ANA_TRE_GIO TABLE	
FIELD NAME	DESCRIPTION
DAT_TRE	Train Date
NUM_TRE	Train Number
SUF_TRE	Train Suffix
NUM_PRC	Train Route Number
COD_PER_FRO	Railway Period Code
PRG_PER_FRO	Railway Period Number
COD_CLS_TTF	Train Classification Code
COD_PDT_TTF	Product Code
COD_DRE_MKT	Marketing Directrix Code
COD_REL_MKT	Marketing Relation Code
COD_STA_INI	Train Leaving Station Code
COD_STA_FIN	Train Final Station Code
TRE_KMT_RET	Railway Kilometre Train

Fig. 5. Train Registry (EDW).

CMM_EDW_TPF_GIO TABLE	
FIELD NAME	DESCRIPTION
DAT_TRE	Train Date
NUM_TRE	Train Number
RET_TRE	Train Railway
COD_RET_INI_TRA	Elementary Route Booking Leaving Railway Code
COD_STA_INI_TRA	Elementary Route Booking Leaving Station Code
COD_RET_FIN_TRA	Elementary Route Booking Final Railway Code
COD_STA_FIN_TRA	Elementary Route Booking Final Station Code
PSI_PRI_PRE	First Class Seats Booked Total Number in the S/T
PSI_SEC_PRE	Second Class Seats Booked Total Number in the S/T
CUC_PRI_PRE	First Class Couchettes Booked Total Number in the S/T
CUC_SEC_PRE	Second Class Couchettes Booked Total Number in the S/T
CUC_CMF_PRE	Total Couchettes Booked Total Number in the S/T
PSI_LET_PRE	Beds Booked Total Number in the S/T
QTA_PSI_TOT_PRE	Booked Travels Total in the S/T
PSI_TAA_PRE	Booked Following Cars Total Number in the S/T
TIP_PRE	Booking Type

Fig. 6. TPF and TPF-120 booking tables (EDW). (TPF-120 table has the DAT_RGT field also.)

that if the booking refers to an associated train leaving after midnight, then it will be brought to the master train of the previous day. In this case, leaving date *x* of the associated will be re-enforced on the *x*+1 date. Both files have the schema shown in Figure 6, except for the TPF-120 table which also contains the DAT_RGT (Registration Date) field. Moreover, the TPF-120 table has the *Daily Delta* field which contains the difference between registered booking data count for each service in the date *r*+1 and that in the date *r* when *p* and *n* are equal (*r* is the record registration date, *p* the train leaving date, and *n* the train number in question). This algorithm enables us to estimate the bookings of the date *r*.

The booking pre-calculation previously described will be done during the ODS to EDW loading phase. In the ODS layer, the structure is the same as the source system's one, and the data calculated will be mixed, for each elementary route, with those deducted from the file ZAWARE01.BATCH.DWH.COMMTPF.TTCGIO (Figure 2). Booking will be again re-calculated in order to consider the Direct Services, defined in terms of Antenna Trains in TTC DB in the file ZAWARE01.BATCH.DWH.COMMTPF.TTCNTN (Figure 3).

4.2 Front-end solutions

Data analyses with OLAP and data mining techniques are used to achieve reports and responses to complex queries. The user (the Marketing Division) accesses to DWHPAX data warehouse in order to accomplish marketing analysis.

Customer's needs about the TPF booking and TPF-120 booking components and about reporting are exemplified in the outputs summarized in the subsequent

figures provided by the Passengers Division Marketing Structure.

In Tables 1 and 2, there are reported the bookings monthly trend data relative to the issuing and leaving dates, respectively, in the four-month period Jan-Apr 2003 and Jan-Apr 2004 with the *monthly delta* variation percent (2004 *vs* 2003). Data are given in thousands.

The crucial component of decisional activity is the summary data section in which a set of statistical indicators regarding the organization data warehouse gives information on the progress of the specific field for which it has been planned.

The analysis indicators are: Time, Train, Product, TTF Classification, Service Type, Service, Directrix, Relation, and S/T.

The metrics that the analysis and reporting phases use, which are self-explaining, are: Railway Km, Train×Km, Delivered seats × Km, Booked travels × Km, Medium Charge, Load Factor *LF*, Booking monitoring, and Daily Delta, where

$$LF = (Booked\_travels \times Km) / (Delivered\_seats \times Km).$$

Reports can be analyzed by the user by a component of the MicroStrategy platform (http://www.microstrategy.com): the MicroStrategy Web front-end, which is accessed with a common Web browser independent of the running operating system. Moreover, it presents an intuitive interface which allows the decision maker to run Business Intelligence applications (*see,* Figure 7).

Examples of reports provided by analytic applications and their graphical representations are shown in Figures 8 to 11. In particular, in Figure 8, the bookings daily data trend is compared to that of the corresponding days of the previous year, while Figure 9 presents such trends to the user graphically. In Figure 10, the table of the bookings monthly trend relative to Jan-Feb 2005 *vs* Jan-Feb 2004, and the corresponding graph are shown. Finally, Figure 11 displays the curve of the 120 days booked seats relative to Dec.'04- Feb.'05 compared to the same period of the previous year.

	Issuing Date		
Month	**2003**	**2004**	**Delta %**
J	1.696	1.893	12%
F	2.183	2.017	-8%
M	2.635	2.426	-8%
A	962	988	-2%
TOT	**7.476**	**7.324**	**-2%**
AVG	**1.869**	**1.831**	

Tab. 1. Bookings monthly trend (issuing date).

	Leaving Date		
Month	**2003**	**2004**	**Delta %**
J	2.423	2.367	-2%
F	2.140	1.977	-8%
M	2.689	2.475	-8%
A	1.255	1.229	-2%
TOT	**8.507**	**8.048**	**-5%**
AVG	**2.126**	**2.012**	

Tab. 2. Bookings monthly trend (leaving date).

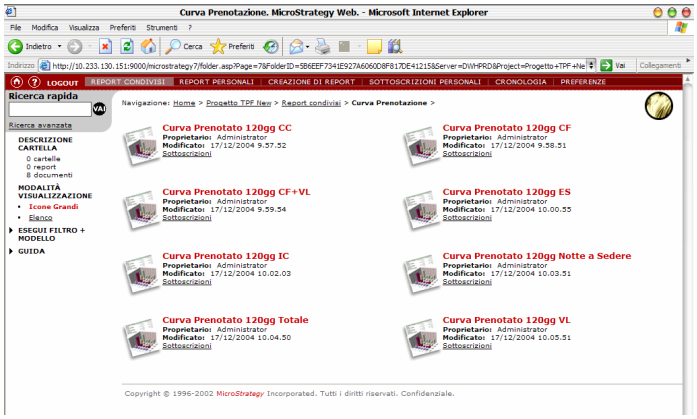
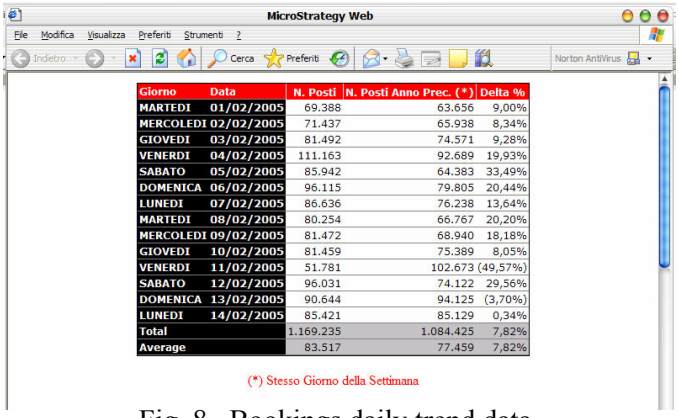


Fig. 7. MicroStrategy Web: user interface.



Giorno	Data	N. Posti	N. Posti Anno Prec. (*)	Delta %
MARTEDI	01/02/2005	69.388	63.656	9,00%
MERCOLEDI	02/02/2005	71.437	65.938	8,34%
GIOVEDI	03/02/2005	81.492	74.571	9,28%
VENERDI	04/02/2005	111.163	92.689	19,93%
SABATO	05/02/2005	85.942	64.383	33,49%
DOMENICA	06/02/2005	96.115	79.805	20,44%
LUNEDI	07/02/2005	86.636	76.238	13,64%
MARTEDI	08/02/2005	80.254	66.767	20,20%
MERCOLEDI	09/02/2005	81.472	68.940	18,18%
GIOVEDI	10/02/2005	81.459	75.389	8,05%
VENERDI	11/02/2005	51.781	102.673	(49,57%)
SABATO	12/02/2005	96.031	74.122	29,56%
DOMENICA	13/02/2005	90.644	94.125	(3,70%)
LUNEDI	14/02/2005	85.421	85.129	0,34%
Total		1.169.235	1.084.425	7,82%
Average		83.517	77.459	7,82%

(*) Stesso Giorno della Settimana

Fig. 8. Bookings daily trend data.

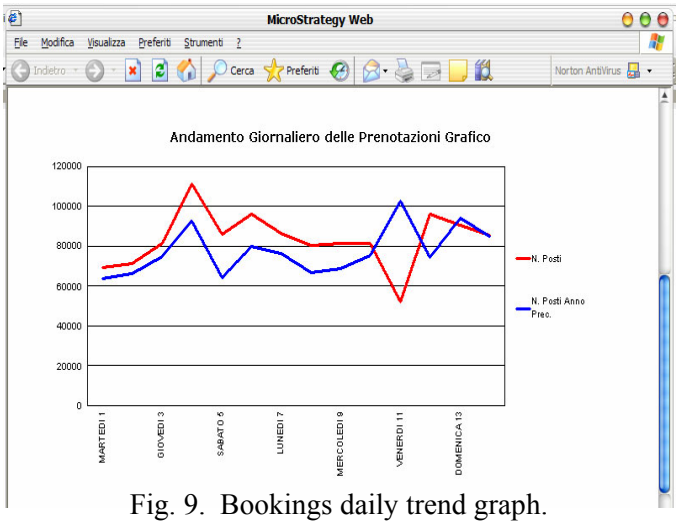


Fig. 9. Bookings daily trend graph.

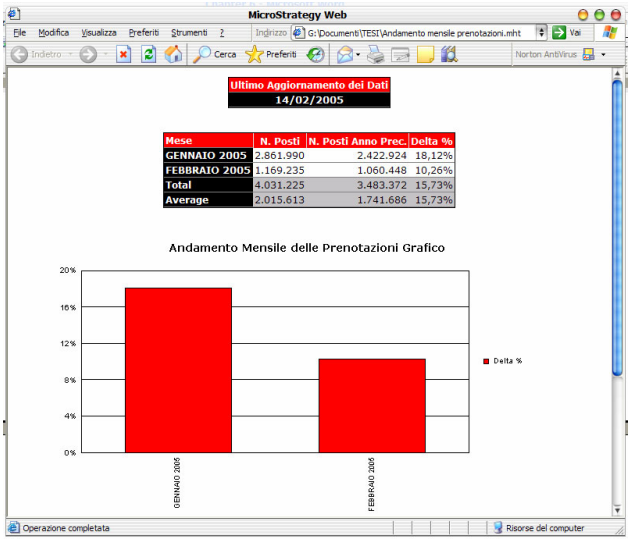


Fig. 10. Bookings monthly trend.

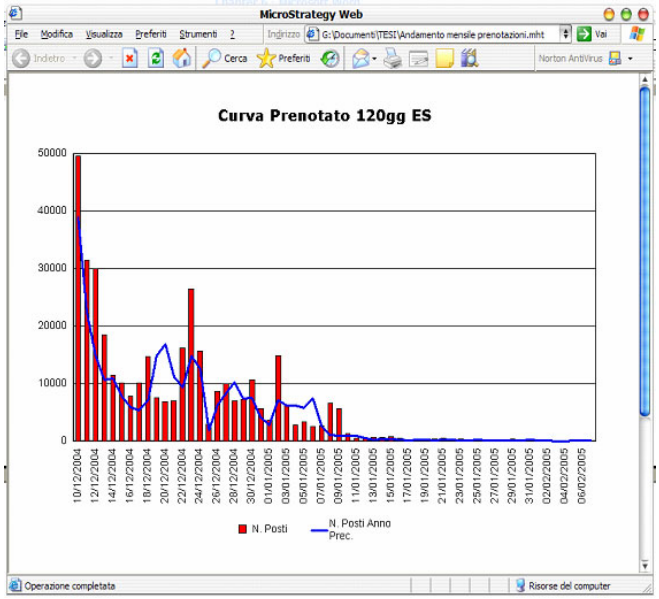


Fig. 11. Bookings four months trend.

5 Concluding remarks

The front-end and back-end solutions for a successful data warehouse have been presented. Data warehousing is useful for organizations producing or manipulating large or huge amounts of data that need to be analyzed. In fact, it provides *ad hoc* analysis tools. Nevertheless, organizations attempting to embrace data warehousing must have a high IT maturity degree, because of the costs that warehouse maintenance implies in terms of constant optimization due to frequent changes of user's needs. An organization, still working to meet its operational information needs, should not embrace data warehousing: the risk for such an organization is the increase of maintaining and re-designing cases.

Finally, it must be said that the choice of an appropriate software, though important, does not provide the guarantee of a successful data warehouse; in fact, it is neither one, nor a combination of software products, but an architecture of system components. For example, the Staging Area, the ETL tools, the Warehouse, the Data Marts, the OLAP, and other tools for end-users can be implemented by different systems.

References

[1] R. Baylis, K. Rich, and J. Fee, *Oracle 9i Database Administrator's Guide*, Release 1 (9.0.1), Oracle Corporation 2001.

[2] M. Boehnlein and A. Ulbrich-vom Ende, Deriving Initial Data Warehouse Structures from Conceptual Data Models of the Underlying Operational Information Systems, *DOLAP '99* Proc. *ACM,* pp. 15-21.

[3] S. Chaundhuri, U. Dayal, and V. Ganti, Database technology for decision support systems, *IEEE Computer*, Vol. 34, No 12, 2001, pp. 48-55.

[4] C. P. Chua and R. Green, *Data Warehousing Fundamentals*, Oracle Corporation 1999.

[5] L. Cupertino, *Building a Successful Data Warehouse: Design, Implementation, and Tuning*, Thesis dissertation 2005.

[6] M. Cyran and C. Dialeris Green, *Oracle 9i Database Performance Guide and Reference*, Release 1 (9.0.1), Oracle Corporation 2001.

[7] C. dell'Aquila, E. Lefons, and F. Tangorra, Decisional portal using approximate query processing, *WSEAS Transactions on Computers*, Vol. 2, No 2, 2003, pp. 486-492.

[8] C. dell'Aquila, E. Lefons, and F. Tangorra, Approximate query processing in decision support system environment, *WSEAS Transactions on Computers*, Vol. 3, No 3, 2004, pp. 581-586.

[9] Gupta A., Harinarayan V., and Quass D., Aggregate-Query Processing in Data Warehousing Environments, *Proc. of the 21st VLDB Conf.,* 1995, pp. 358-369.

[10] M. Janesch, *Implementing the Best Data Warehousing Tuning Techniques for Your Environment*, Innovative Consulting 2001.

[11] M. Jarke, M. Lenzerini, Y. Vassiliou, and P. Vassiliadis, *Fundamentals of Data Warehouses*, Springer-Verlag, 2003.

[12] P. Lane, V. Shupmann, *Oracle 9i Warehousing Guide*, Release 1 (9.0.1), Oracle Corporation 2001

[13] L. McGeen Lusher, *Oracle 9i Database Concepts*, Release 1 (9.0.1), Oracle Corporation 2001.

[14] H. Ong, *Data Warehouse Myths and Misconceptions*, Aurora Consulting 1999.